APIM

<<Physical>>

BCM

<<Physical>>

Classic Memory Physical

<<Physical>>

DDM

<<Physical>>

DSM

<<Physical>>

ECG

<<Physical>>

ECM

<<Physical>>

GFM2\_SCSM

<<Physical>>

GWM\_ECG

<<Physical>>

HPCM

<<Physical>>

HUD

<<Physical>>

IPC

<<Physical>>

PDM

<<Physical>>

RCM

<<Physical>>

SCMB\_PSM

<<Physical>>

SCMG/SCMH

<<Physical>>

SOBDMC\_HPCM

<<Physical>>

|  |  |  |  |
| --- | --- | --- | --- |
| Document Type | **Feature Implementation Specification (FIS)** | |  |
| Template Version | **6.1a** | |  |
| SysML Report Template Version | **O Beta (11/24/2019)** | |  |
| Document ID | **ffst01.10\_featuredocument\_sysmlreporttemplate** | |  |
| Document Location |  | |  |
| Document Owner |  | |  |
| Document Revision | **FD0** | |  |
| Document Status | **Draft** | |  |
| Date Issued | **2021/03/08** | |  |
| Date Revised | **2021/03/08** | |  |
| Document Classification | GIS1 Item Number: | **27.60/35** |  |
| GIS2 Classification: | **Confidential** |

|  |  |  |  |
| --- | --- | --- | --- |
| Document Approval | | | |
| Person | Role | Email | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

This document contains Ford Motor Company Confidential information. Disclosure of the information contained in any portion of this document is not permitted without the expressed, written consent of a duly authorized representative of Ford Motor Company, Dearborn, Michigan, U.S.A.

Copyright © 2019, Ford Motor Company

PRINTED COPIES ARE UNCONTROLLED

**Disclaimer**

**This document contains Ford Motor Company Confidential information. Disclosure of the information contained in any portion of this document is not permitted without the expressed, written consent of a duly authorized representative of Ford Motor Company, Dearborn, Michigan, U.S.A.**

This document contains information developed and accumulated by and for FORD MOTOR COMPANY. As such, it is a proprietary document, which, if disseminated to unauthorized persons, would provide others with restricted information, data, or procedures not otherwise available, exposing the FORD MOTOR COMPANY to potential harm.

Employees and suppliers having custody of this specification or authorized to use it must be cognizant of its proprietary nature and ensure that the information herein is not made available to unauthorized persons.

FORD MOTOR COMPANY reserves the right to protect this work as an unpublished copyrighted work in the event of an inadvertent or deliberate unauthorized publication. FORD MOTOR COMPANY also reserves its rights under copyright laws to protect this work as a published work.

This document or portions thereof shall not be distributed outside FORD MOTOR COMPANY without prior written consent. Refer all questions concerning disclosure to the author(s) or any duly authorized representative of Ford Motor Company.

**Copyright** © **2019 Ford Motor Company**

**Contents**

1 Introduction 6

1.1 Document Purpose 6

1.2 Document Scope 6

1.3 Document Audience 6

1.3.1 Stakeholder List 6

1.4 Document Organization 6

1.4.1 Document Context 6

1.4.2 Document Structure 6

1.5 Document Conventions 7

1.5.1 Requirements Templates 7

1.5.1.1 Identification of requirements 7

1.5.1.2 Requirement Attributes 7

2 Feature Implementation Overview 8

2.1 Description 8

2.2 Input Requirements 8

2.3 Assumptions 9

2.4 References 9

2.4.1 Ford Documents 9

2.4.2 External Documents and Publications 10

2.5 Glossary 10

2.5.1 Definitions 10

2.5.2 Abbreviations 11

3 Feature Implementation Architecture 12

3.1 Functional Architecture 12

3.1.1 Description 12

3.1.2 Function List 12

3.1.2.1 Functions of 13

3.1.2.2 Functions of 14

3.1.3 Signal List 14

3.2 Physical Architecture 18

3.2.1 E/E Architecture 18

3.2.1.1 E/E Architecture Variants 19

3.2.1.2 E/E Components 20

3.2.1.3 E/E Connections 20

3.2.1.4 Signal List 22

3.2.2 Software Component Architecture 22

3.2.2.1 Description 22

3.3 Function Deployment 22

3.3.1 Deployment Variants 23

3.3.1.1 Main (Only) variant 23

3.3.1.2 Deployment “Variant 1” 23

3.3.2 Function Allocation 24

3.3.2.1 Functional Safety 25

4 Feature Implementation Modeling 27

4.1 Component Interaction Diagrams 27

4.1.1 Scenario: “System Startup / Shutdown” 27

4.1.2 Scenario: “Normal Operation” 27

4.1.3 Functional Safety 28

4.1.3.1 Fault Handling Time Analysis 28

4.1.3.2 Requirements Derivation Diagram 29

4.2 Component Interface Behavior Diagrams 29

5 Feature Implementation Requirements 30

6 Open Concerns 31

7 Revision History 32

8 Appendix 33

8.1 Data Dictionary 33

8.1.1 Logical Signals 33

8.1.2 Logical Parameters 33

8.1.3 Technical Signals 33

8.1.3.1 GSDB Signals 34

8.1.3.2 Service Oriented Communication 34

8.1.3.3 Hardwired Signals 34

8.1.3.4 Diagnostic Interfaces 34

8.1.4 Technical Parameters 36

8.1.5 Mappings 36

8.1.6 Technical Interfaces 37

8.1.6.1 AIS Interfaces 37

8.1.6.2 Service Oriented Communcation (SoC) Interfaces 38

8.1.6.3 AUTOSAR Ports (SW Interfaces) 38

8.1.7 Messages 38

8.1.7.1 CAN Bus “<Bus Name>” 38

8.1.7.2 LIN Bus “<Bus Name>” 39

8.1.8 Encoding Types 39

**List of Figures**

Figure 1: 21

Figure 2:  24

Figure 3: 30

**List of Tables**

Table 1‑1: Electrical Architecture(s) referenced in this document 6

Table 1‑2: Functions referenced in this document 6

Table 2‑1: Ford Internal Documents 10

Table 2‑2: Ford Internal Documents *(not specified in SysML model)* 11

Table 2‑3: External Documents and Publications 11

Table 2‑4: External Documents and Publications *(not specified in SysML model)* 11

Table 2‑5: Definitions Used In This Document 12

Table 2‑6: Abbreviations used in this document 12

Table 3‑1: List of Functions 15

Table 3‑2: List of ‑‑ Functions 15

Table 3‑3 List of signals sent by 18

Table 3‑4 List of signals received by 19

Table 3‑5: Electrical Components 21

Table 3‑6: E/E Connections for 23

Table 3‑7: Function Allocation Table 26

Table 3‑8: Function Allocation Table 27

Table 3‑9: Architectural Redundancy Summary 27

Table 4‑1: Fault Handling Time Table 30

Table 5‑1: Input Signal mappings of ‑ 34

Table 5‑2: Output Signal mappings of ‑ 35

Table 5‑3: Parameter mappings of ‑ 36

Table 5‑4: Component Specific Requirements 37

Table 5‑5: Inherited Requirements 37

Table 5‑6: Input Signal mappings of ‑ 39

Table 5‑7: Output Signal mappings of ‑ 40

Table 5‑8: Parameter mappings of ‑ 40

Table 5‑9: Component Specific Requirements 40

Table 5‑10: Inherited Requirements 40

Table 5‑11: Input Signal mappings of Component: ‑ 41

Table 5‑12: Output Signal mappings of Component: ‑ 42

# Introduction

## Document Purpose

The Feature Implementation Specification (FIS) specifies the deployment of the logical functions of a feature to an electrical architecture. The FIS specifies all interactions between the ECUs of the electrical architecture required for the feature including the technical signals and the interfaces. It also gives interface and integration requirements, which are specific to the feature for the electrical architecture.

To get more information about the concept of feature, function and component level abstraction refer to the [Ford RE Wiki](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Engineering+for+SW+Enabled+Features).

## Document Scope

**#Hint:** The FIS can be used to document multiple deployment variants (refer to chapters “Deployment Variants” and “E/E Architecture Variants”). It is however recommended (except for small features) to have a separate FIS for each variant, because managing multiple variants in the same document easily gets complex and cumbersome.

**#Functional Safety:** For Functional Safety specify only one deployment variant per FIS.

This FIS describes the deployment of the feature <Feature> to the following electrical architecture(s):

| **Electrical Architecture Name** | **Owner** | **Reference** |
| --- | --- | --- |
| e.g. CGEA1.3 |  | <Add VSEM Link> |
|  |  |  |

Table 1‑1: Electrical Architecture(s) referenced in this document

## Document Audience

The FIS is authored by - . All Stakeholders, i.e., all people who have a valid interest in the feature implementation should read and, if possible, review the FIS. It needs to be guaranteed, that all stakeholders have access to the currently valid version of the FIS.

**#Hint:** The FIS template has the IP Classification “Proprietary” by default. IP Classification “Confidential” might be required in some cases, e.g. by Ford Functional Safety.

**#Macro:** [Add Ins -> Edit Document Properties macro](http://wiki.ford.com/display/RequirementsEngineering/Editing+Specification+Properties)

### Stakeholder List

For the latest stakeholder list of the feature and their influence refer to [VSEM ID: VDOC087225](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=Rza9MpzOx3NrTDAAAAAAAAAAAAA&servername=Production_Server).

## Document Organization

### Document Context

Refer to the [Specification Structure page](http://wiki.ford.com/display/RequirementsEngineering/Specification+templates) in the [Ford RE Wiki](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Engineering+for+SW+Enabled+Features) to understand how the FIS relates to other Ford Requirements Documents and Specifications.

### Document Structure

The structure of this document is explained below:

**Section 1** – Introduction – Giving an explanation how to use this document including responsibilities and the scope of the document. Additionally it contains the revision history and a list of unsettled but known issues that have to be consolidated in future versions. It explains the terminology and gives a clarification of the definitions, concepts and abbreviations used in the document.

**Section 2** – Feature Implementation Description – Giving an overview of the platform and listing assumptions, constraints or dependencies

**Section 3** – Feature Implementation Architecture – Describing 3 Architecture Views:

* Functional Architecture – Showing the logical architecture of functions
* Physical Architecture – Showing the physical architecture (first of all the E/E Architecture), which the Logical Functions get allocated to.
* Software Architecture – Showing the software architecture relevant for the feature (for features with in-house development only)
* Function Deployment – Presenting the allocation of logical functions and signals to the electrical and other components

**Section 4** – Deployment Specific Modeling –Modeling techniques providing additional detail on e.g. interface behavior

**Section 5** – Deployment Specific Requirements – Deployment specific requirements for ECUs, Network Communication, and Process

**Section 6** – List of Open Concerns

**Section 7** – Revision History

**Section 8** – Appendix - Presenting additional data mainly in a tabular form, e.g., a data dictionary

## Document Conventions

### Requirements Templates

Refer to “[How to use the Specification Templates](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates?src=contextnavpagetreemode)” on how to use the specification templates and the VBA macros to create/edit the requirements in the specifications.

The VBA macro enable the import of the specification to VSEM (refer to ["How to import specifications into VSEM as separate requirements"](http://wiki.ford.com/pages/viewpage.action?pageId=104991616&src=contextnavpagetreemode)).

#### Identification of requirements

The unique requirement ID given in the headline of any requirement follows the requirement throughout the development process. The requirement ID format follows a well-defined syntax.

All identifiers in an FIS shall be composed of 4 parts:

* A leading prefix, which indicates the type of requirement (R=Requirement, UC=Use Case, SC=Scenario, …)
* A prefix, which indicates the abstraction level (F=Feature, FNC=Function, CMP = component).
* Followed by a name, indicating the scope, which the requirement belongs to (e.g. feature or function name )
* Ending with the actual requirement number

*Example:*

*R\_CMP\_LockArbitrator\_00004* This is the fourth requirement on component level for the function Lock Arbitrator.

#### Requirements Attributes

Additionally attributes can be added to each requirement. This helps to classify requirements. A [list of available attributes](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes?src=contextnavpagetreemode) is given in the RE Wiki.

## References

### Ford Documents

The list of all Ford internal documents, which are directly related.

### External Documents and Publications

The list of external documents could include books, reports and online sources.

**#Hint:** You may refer to [IEEE Citation Reference](http://www.ieee.org/documents/ieeecitationref.pdf) on how to format a reference.

## Glossary

### Definitions

| **Definition** | **Description** |
| --- | --- |
| Chime | An audible alert |
| Classic Memory Recall | When Classic Memory performs a recall of positional settings. Positoinal settings are recalled from a profile. Note: other features perform recalls. |
| Classic Memory Store | When Classic Memory performs a store of positional settings. Positoinal settings are stored to a profile. Note: other features perform stores. |
| Configurable Parameter | The setting is a variable that can be adjusted by a technician |
| DCO | Effort to remove the key slot/PEPS from vehicle. Drives changes to other features that relied on the key slot/PEPS for input triggers. |
| Drive Position | The position where the Driver has manually set positional settings. The assumption is this is the preferred position from which the Driver plans to drive the vehicle. |
| EEEE Offset Position | This position has the seat rearward and the steering column forward to allow more space for entering and exiting the vehicle. |
| Memory Commodity | A system that has adjustable positions and memory capability (ex. Memory Seat can store and recall positions to memory) |
| Memory Input | The interface which a user can use to select a Profile to recall or store. |
| Memory Location | The location which stores the settings for a profile |
| Memory Set | A variant of Memory Input has Memory Set and profile options (instead of just profile options). Instead of pressing and holding a profile option to perform a memory store, this variant has the user select Memory Set then the profile option to perform a memory store. Note: Pressing a memory option in the variant performs a memory recall as normal. |
| Multicontour Seat | A specialized seat that contains adjustable massagers and air bladders |
| Personality | Another term for ‘Profile’. Terms have changed over the years and may not align with signals. |
| Profile | A profile stores the settings for a user |
| Prompt | A visual alert |
| Vehicle Speed Threshold | The speed at which it is determined to be too dangerous to trigger a positional setting recall. |
| Wait Mode | When Classic Memory is expecting input of a certain type. When Wait Mode is entered it will specify which input will trigger its exit. |

Table 1‑4: Definitions used in this document

### Abbreviations

| **Abbr.** | **Stands for** | **Description** |
| --- | --- | --- |
| ACM | Advanced Classic Memory |  |
| AHUD | Advanced Head-Up Display |  |
| APIM | Accessory Protocol Interface Module |  |
| BCM | Body Control Module |  |
| CM | Classic Memory |  |
| DCO | Drive Control Optimization |  |
| DDM | Driver Door Module |  |
| DSM | Drivers Seat Module |  |
| ECG | Enhanced Central Gateway |  |
| ECM | Engine Control Module |  |
| EEEE | Easy Entry / Easy Exit |  |
| EM | Enhanced Memory |  |
| HMI | Human-Machine Interface |  |
| HUD | Head-Up Display |  |
| IPC | Instrument Panel Cluster |  |
| PDCM | Primary Drive Control Module |  |
| PDM | Passenger Door Module |  |
| PPP | Personal and Portable Profile |  |
| PSM | Passenger Seat Module |  |
| RCM | Restraint Control Module |  |
| RKE | Remote Key-less Entry |  |
| SCMG | Seat Control Module G | Multicontour Seat |
| SCMH | Seat Control Module H | Multicontour Seat |
| SCSM | Steering Column Switch Module |  |
| SSW | Stowable Steering Wheel |  |

Table 1‑5: Abbreviations used in this document.

# Feature Implementation Overview

## Description

APIM

* Contains Enhanced Memory feature
* Prompt user
* Chime user

BCM

* Arbitrates recalls
* PaaK status
* Door Ajar status

DDM

* Side Mirror abitration

DSM

* Classic Memory arbitration (including Easy Entry Esy Exit)

ECG

* Contains Auto Save system

RCM

* Seatbelt status

SCMG/SCMH

* Multicontour seat

SOBDMC\_HPCM

* Park status

## Input Requirements/Documents

No Input

## Lessons Learned

No lessons learned specified.

## Assumptions

No Assumptions specified.

# Feature Implementation Architecture

## Functional Architecture

### Description

This diagram contains all the functions required to Store and Recall the positional settings. This includes inputs and outputs to the user, other features, and external systems.

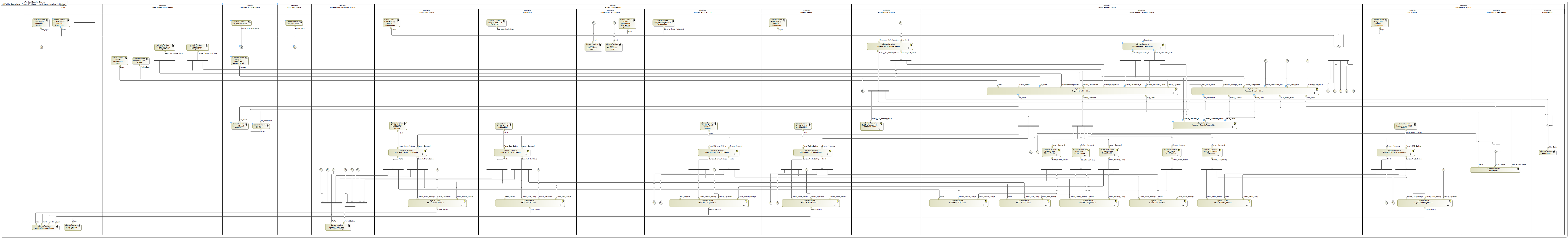


Figure 8: Classic Memory Functional Architecture

This diagram contains all the functions required to adjust seats and steering to provide additional space for the user to enter and exit. This includes inputs and outputs to the user, other features, and external systems.

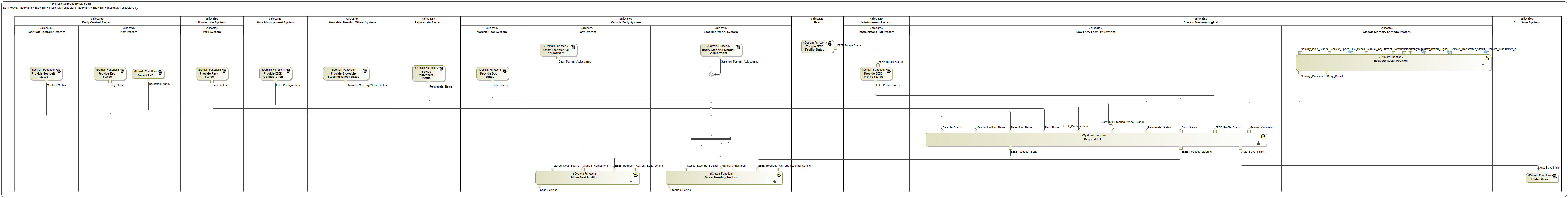


Figure 9: Easy Entry Easy Exit Functional Architecture

### Function List

The following functions from the [Global Feature & Function List](https://www.vsemweb.ford.com:443/tc/launchapp?-attach=true&-s=226TCSession&-o=ZmZNi0JHx3NrTDAAAAAAAAAAAAA) are referenced in this Feature Implementation Specification:

| **Function ID** | Function Name | Function Description |
| --- | --- | --- |
|  | *(activity)* Move Seat Position | *(activity)* The purpose of this function is to adjust the position of the seats to a stored value. This is the final step of the Classic Memory recall process. This function receives the stored settings and applies them. |
|  | *(activity)* Request Recall Settings | *(activity)* (activity) The purpose of this function is to indicate to Enhanced Memory when Classic Memory performed a recall. Enhanced Memory will then respond by performing a recall as well. |
|  | *(activity)* Provide Transmission Status | *(activity)* (activity) The purpose of this function is to indicate the status of the transmission (gear). |
|  | *(activity)* Provide Restriction Settings Status | *(activity)* (activity) The purpose of this function is to indicate the status of the Restriction Setting Status. |
|  | *(activity)* Provide Feature Configuration | *(activity)* activity) The purpose of this function is to indicate if the feature variant is implemented. |
|  | *(activity)* Provide Actual Pedals Settings | *(activity)* (activity) The purpose of this function is to indicate the current values of the pedals. |
|  | *(activity)* Detect Remote Transmitter | *(activity)* activity) The purpose of this function is to detect when a remote transmitter sends out a Lock or Unlock signal. This function then indicates what signal was received and by which remote transmitter. Other functions will use this data to determine if a recall should occur or if a profile should add/remove a remote transmitter. |
|  | *(activity)* Store Seat Position | *(activity)* The purpose of this function is to store the position of the seats to a profile. This is the final step of the Classic Memory store process. This function receives the current settings and applies them to the user selected profile. |
|  | *(activity)* Provide Actual Side Mirrors Settings | *(activity)* The purpose of this function is to indicate the current values of the side mirrors. |
|  | *(activity)* Provide Actual Seat Settings | *(activity)* (activity) The purpose of this function is to indicate the current values of the seat. |
|  | *(activity)* Associate Remote Transmitter | *(activity)* (activity) The purpose of this function is to add or remove a remote transmitter from a profile. |
|  | *(activity)* Read AHUD Stored Brightness | *(activity)* The purpose of this function is to obtain the stored value of the AHUD brightness. The settings will be applied to the current AHUD brightness. |
|  | *(activity)* Provide Actual AHUD Settings | *(activity)* (activity) The purpose of this function is to indicate the current value of the AHUD brightness. |
|  | *(activity)* Update Profile with Positional Settings | *(activity)* (activity) The purpose of this function is to update the Personal and Portable Profile’s profile. When Classic Memory updates a profile setting then this function makes the same update. |
|  | *(activity)* Store AHUD Brightness | *(activity)* The purpose of this function is to store the brightness of the AHUD to a profile. This is the final step of the Classic Memory store process. This function receives the current settings and applies them to the user selected profile. |
|  | *(activity)* Notify Multicontour Seat Manual Adjustment | *(activity)* (activity) The purpose of this function is to indicate when the Multicontour Seat has made a manual adjustment. Other functions use this data to determine when to cancel automatic movement. |
|  | *(activity)* Recall Multicontour Seat | *(activity)* (activity) The purpose of this function is to automatically adjust the positional settings of the Multicontour Seat. |
|  | *(activity)* Provide Vehicle Speed | *(activity)* (activity) The purpose of this function is to indicate the current speed of the vehicle. |
|  | *(activity)* Provide Memory Input Status | *(activity)* The purpose of this function is to indicates when and which input the user selects. This function processes some of the input depending on the type of memory input available (Memory Set). |
|  | *(activity)* Lock/Unlock via Remote Transmitter | *(activity)* (activity) The purpose of this function is to indicate when the user has triggered either Lock or Unlock signal via a remote transmitter. |
|  | *(activity)* Notify Seat Manual Adjustment | *(activity)* (activity) The purpose of this function is to indicate when the Seat has made a manual adjustment. Other functions use this data to determine when to cancel automatic movement. |
|  | *(activity)* Read Seat Current Position | *(activity)* The purpose of this function is to obtain the current value of the seat positions. This function also identifies a profile that settings should be stored to. |
|  | *(activity)* Notify of Enhanced Memory Recall | *(activity)* (activity) The purpose of this function is to indicate when the Enhanced Memory feature has performed a recall. The Classic Memory feature’s functions will respond by performing a recall as well. |
|  | *(activity)* Notify of Memory Set Indicator Status | *(activity)* The purpose of this function is to alert the user that Memory Set is active. The user has activated the Memory Set functionality and this function will notify the user for the duration that it is active. |
|  | *(activity)* Adjust AHUD Brightness | *(activity)* The purpose of this function is to adjust the brightness of the AHUD to a stored value. This is the final step of the Classic Memory recall process. This function receives the stored settings and applies them. |
|  | *(activity)* Read Mirrors Current Position | *(activity)* The purpose of this function is to obtain the current value of the mirror positions. This function also identifies a profile that the settings should be stored to. |
|  | *(activity)* Store Pedals Position | *(activity)* The purpose of this function is to store the position of the pedals to a profile. This is the final step of the Classic Memory store process. This function receives the current settings and applies them to the user selected profile. |
|  | *(activity)* Read Pedals Stored Position | *(activity)* The purpose of this function is to obtain the stored value of the pedal positions. The settings will be applied to the current pedal positions. |
|  | *(activity)* Notify Audio | *(activity)* (activity) The purpose of this function is to notify the user via an audible alert. |
|  | *(activity)* Store Steering Position | *(activity)* The purpose of this function is to store the position of the steering to a profile. This is the final step of the Classic Memory store process. This function receives the current settings and applies them to the user selected profile. |
|  | *(activity)* Read Seat Stored Position | *(activity)* The purpose of this function is to obtain the stored value of the seat positions. The settings will be applied to the current seat positions. |
|  | *(activity)* Read Mirrors Stored Position | *(activity)* The purpose of this function is to obtain the stored value of the mirror positions. The settings will be applied to the current mirror positions. |
|  | *(activity)* Receive Visual Status | *(activity)* (activity) The purpose of this function is to indicate that the user is aware of the visual outputs. |
|  | *(activity)* Notify Mirrors Manual Adjustement | *(activity)* (activity) The purpose of this function is to indicate when a side mirror has made a manual adjustment. Other functions use this data to determine when to cancel automatic movement. |
|  | *(activity)* Notify Steering Manual Adjustment | *(activity)* (activity) The purpose of this function is to indicate when the steering column has made a manual adjustment. Other functions use this data to determine when to cancel automatic movement. |
|  | *(activity)* Read Steering Stored Position | *(activity)* The purpose of this function is to obtain the stored value of the steering positions. The settings will be applied to the current steering positions. |
|  | *(activity)* Store/Recall Positional Settings | *(activity)* (activity) The purpose of this function is to indicate when the user has triggered either a recall or store via the Memory Input. |
|  | *(activity)* Receive Positional Status | *(activity)* (activity) The purpose of this function is to indicate that the user is aware of the positional setting adjustments. |
|  | *(activity)* Store Mirrors Position | *(activity)* The purpose of this function is to store the position of the mirrors to a profile. This is the final step of the Classic Memory store process. This function receives the current settings and applies them to the user selected profile. |
|  | *(activity)* Request Store Position | *(activity)* The purpose of this function is to determine that the result of the user input is a store action. From this determination this function will then trigger a Classic Memory store. This store will retrieve current settings and store them to the user select profile. |
|  | *(activity)* Notify Pedals Manual Adjustment | *(activity)* (activity) The purpose of this function is to indicate when a pedal has made a manual adjustment. Other functions use this data to determine when to cancel automatic movement. |
|  | *(activity)* Read AHUD Current Brightness | *(activity)* The purpose of this function is to obtain the current value of the AHUD brightness. This function also identifies a profile that the settings should be stored to. |
|  | *(activity)* Move Pedals Position | *(activity)* (activity) The purpose of this function is to adjust the position of the pedals to a stored value. This is the final step of the Classic Memory recall process. This function receives the stored settings and applies them. |
|  | *(activity)* Auto Save Store | *(activity)* (activity) The purpose of this function is to indicate when the Auto Save system has determined that the current profile should perform a Classic Memory store. |
|  | *(activity)* Store Multicontour Seat | *(activity)* (activity) The purpose of this function is to store the current Multicontour Seat settings to the indicated profile. |
|  | *(activity)* EM\_Store | *(activity)* (activity) The purpose of this function is to indicate which profile Classic Memory should perform a Classic Memory store. This is part of the new profile creation process. |
|  | *(activity)* Move Mirrors Position | *(activity)* The purpose of this function is to adjust the position of the mirrors to a stored value. This is the final step of the Classic Memory recall process. This function receives the stored settings and applies them. |
|  | *(activity)* Read Steering Current Position | *(activity)* The purpose of this function is to obtain the current value of the steering positions. This function also identifies a profile that settings should be stored to. |
|  | *(activity)* Display HMI | *(activity)* (activity) The purpose of this function is to notify the user via a visual alert. |
|  | *(activity)* Move Steering Position | *(activity)* The purpose of this function is to adjust the position of the steering column to a stored value. This is the final step of the Classic Memory recall process. This function receives the stored settings and applies them. |
|  | *(activity)* Notify AHUD Manual Adjustment | *(activity)* (activity) The purpose of this function is to indicate when the AHUD has made a manual adjustment. Other functions use this data to determine when to cancel automatic movement. |
|  | *(activity)* Provide Actual Steering Settings | *(activity)* (activity) The purpose of this function is to indicate the current values of the steering column. |
|  | *(activity)* Create New Profile | *(activity)* (activity) The purpose of this function is to indicate when the Enhanced Memory feature has begun the process of creating a new profile for the user. |
|  | *(activity)* Read Pedals Current Position | *(activity)* The purpose of this function is to obtain the current value of the pedal positions. This function also identifies a profile that settings should be stored to. |
|  | *(activity)* Request Recall Position | *(activity)* The purpose of this function is to determine that the result of the user input is a recall action. From this determination this function will then trigger a Classic Memory recall. This recall will retrieve stored settings from a user selected profile and apply them to the current settings. |

Table 17: List of Functions on Classic Memory Functional Architecture

| **Function ID** | Function Name | Function Description |
| --- | --- | --- |
|  | *(activity)* Provide Rejuvenate Status | *(activity)* (activity) The purpose of this function is to indicate when the Rejuvenate feature is active. Easy Entry Easy Exit functions will be restricted when Rejuvenate is active. |
|  | *(activity)* Notify Steering Manual Adjustment | *(activity)* (activity) The purpose of this function is to indicate when the steering column has made a manual adjustment. Other functions use this data to determine when to cancel automatic movement. |
|  | *(activity)* Detect RKE | *(activity)* (activity) The purpose of this function is to detect the RKE. |
|  | *(activity)* Provide Stowable Steering Wheel Status | *(activity)* (activity) The purpose of this function is to indicate when the Stowable Steering Wheel feature is active. Easy Entry Easy Exit functions will be restricted when Stowable Steering Wheel is active. |
|  | *(activity)* Provide EEEE Configuration | *(activity)* (activity) The purpose of this function is to indicate when Drive Control Optimization variant is implemented. |
|  | *(activity)* Inhibit Store | *(activity)* (activity) The purpose of this function is to inhibit Auto Save to do a store |
|  | *(activity)* Move Seat Position | *(activity)* The purpose of this function is to adjust the position of the seats to a stored value. This is the final step of the Classic Memory recall process. This function receives the stored settings and applies them. |
|  | *(activity)* Provide Door Status | *(activity)* (activity) The purpose of this function is to indicate the status of the driver door. |
|  | *(activity)* Request EEEE | *(activity)* The purpose of this function is to determine when the user is entering and exiting the vehicle. From that determination this function will then adjust the seat and steering positions. |
|  | *(activity)* Provide Key Status | *(activity)* (activity) The purpose of this function is to indicate the status of the key slot / push to start (ignition). |
|  | *(activity)* Toggle EEEE Profile Status | *(activity)* (activity) The purpose of this function is to indicate when the user toggles Easy Entry Easy Exit between ON and OFF for a profile. |
|  | *(activity)* Provide Seatbelt Status | *(activity)* (activity) The purpose of this function is to indicate the status of the driver seatbelt. |
|  | *(activity)* Notify Seat Manual Adjustment | *(activity)* (activity) The purpose of this function is to indicate when the Seat has made a manual adjustment. Other functions use this data to determine when to cancel automatic movement. |
|  | *(activity)* Provide Park Status | *(activity)* (activity) The purpose of this function is to indicate when the vehicle is not in motion (Park or Neutral). |
|  | *(activity)* Request Recall Position | *(activity)* The purpose of this function is to determine that the result of the user input is a recall action. From this determination this function will then trigger a Classic Memory recall. This recall will retrieve stored settings from a user selected profile and apply them to the current settings. |
|  | *(activity)* Move Steering Position | *(activity)* The purpose of this function is to adjust the position of the steering column to a stored value. This is the final step of the Classic Memory recall process. This function receives the stored settings and applies them. |
|  | *(activity)* Provide EEEE Profile Status | *(activity)* (activity) The purpose of this function is to indicate if a profile has Easy Entry Easy Exit set ON or OFF. |

Table 17: List of Functions on Easy Entry Easy Exit Functional Architecture

### Signal List

*#Hint: Refer to the Data Dictionary - Logical Signals.*

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Description** | **Details** |
| **Cancel\_Auto\_Movement** | Indicates current automatic movement should cease | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Seat Belt Status** | Indicates status of the driver seat belt | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Gear Status** | Indicates status of Park/Neutral | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **FirstRowBuckleDriver** | Indicates status of driver seatbelt | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Current Setting** | Current Position of the different commodities | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **HMI Request Recall** | Recall request from SYNC (Enhanced Memory) to BCM | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersNoPos\_D\_Actl** | Recall request from BCM to DSM (Classic Memory) | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Chime Status** | Request chime to be audible | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Door Status** | Indicates status of Door | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Park Status** | Indicates if vehicle is in Park | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **SSW\_Active** | Indicate Stowable Steering Wheel feature is active | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory Command** | Indicates Classic Memory request to either store or recall positional settings | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemSwtch\_D\_RqAssoc** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **EmPrflButtnAssoc\_D\_Rq** | Enhanced Memory requesting Classic Memory to Button Association Mode | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Prompt Status** | Request prompt to be displayed | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory Input Status** | Indicates user selected profile | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **TmPrkSys\_D\_Actl** | Indicates Park status | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key\_In\_Ignition\_Stat** | Indicates Key status | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_Cmd** | Indicates to store or recall a profile | Satisfies:  799476293.png Example Implementation Reqt  Source ECU:  Target ECU: |
| **MemSwtch\_D\_RqRecall** | DSM (Classic Memory) requesting recall to BCM | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_[x]\_SwPsngr\_Stat** | Gerneric version signal for Memory Switch status. [x] is the profile number (1-4). | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Easy\_Entry\_Rqst** | Requests automatic movement of EEEE commodities | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Switch Request Recall** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Rejuvenate\_Active** | Indicates if Rejuvenate feature is active | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **DrStatDrv\_B\_Actl** | Indicates driver door status | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request Current Position** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Stowable Steering Wheel Status** | Indicates the status of the Stowable Steering Wheel | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_Feedback\_Rqst** | Request to cluster to chime | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request\_Store** | Request to trigger a store action | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersNo\_D\_Actl** | Requests recall from BCM to APIM (Enahnced Memory) | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Mirror\_Manual\_Override** | Indicates manual adjustment of mirrors | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Strategy Configuration** | Indicates status of Easy Entry Easy Exit | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Restriction Status** | Indicates restriction status of profiles | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersRecallSrc\_D\_Actl** | Indicates the source of a recall | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersStore\_D\_Rq** | APIM (Enhanced Memory) requests DSM (Classic Memory) to store settings to a profile | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Personal Index Value** | Indicates user selected profile | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RecallEvent\_No\_Cnt** | A counter used to align recalls | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemDvrDeny\_B\_Stat** | Request cluster to notify user that a recall is not currently possible | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **CntrStk\_D\_RqRecall** | Recall request from APIM (Enahnced Memory) | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Vehicle Speed** | Indicates the speed of the vehicle | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **User Input** | User selects a memory button/profile | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_[x]\_Sw\_Stat** | Generic signal from DDM to DSM indicating input from Memory Switches | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Manual Adjustment** | Indicates if there has been a manual adjustment of any commodity setting | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **EEEE Profile Status** | Indicates if EEEE is enabled or disabled | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemStoreMsgTxt\_D\_Rq** | Request to cluster to prompt the user | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemSwtchPsngrFdbck\_B\_Rq** | Request to cluster to prompt the user | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key Detection Status** | Indicates the Status of the a Key | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PaakInVehicle** | Indicates if PaaK is in the vehicle | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Button Association Mode** | Indicates status of Button Association Mode | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |

## Physical Architecture

### E/E Architecture

#### E/E Architecture Variants

|  |  |  |
| --- | --- | --- |
| E/E Architecture Variant Name | Variant Description | Variant Condition (optional) |
| No Vairants Defined |  |  |
|  |  |  |
|  |  |  |

##### E/E Architecture: Physical Architecture

This E/E Architecture variant does not have memory switches and triggers based on input from other features (such as Enhanced Memory and Personal and Portable Profiles). Drive Control Optimization also replaces the Key In Ignition Status with new inputs (Park, Seatbelt, Door, and Paak) as triggers to Easy Entry Easy Exit (EEEE).

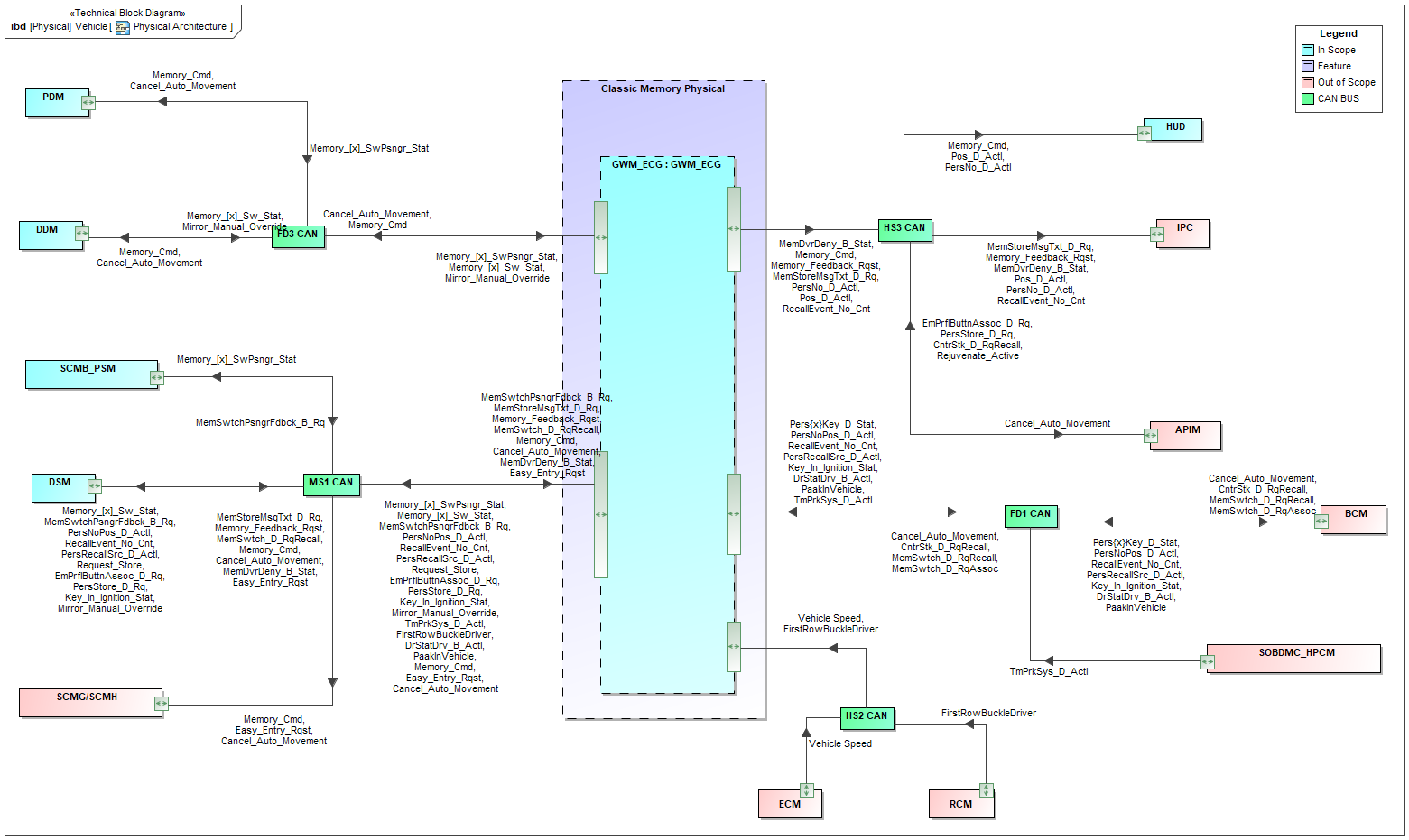


Figure 2: Physical Architecture

#### E/E Components

|  |  |
| --- | --- |
| **Component Name** | **Description** |
| APIM | Accessory Protocol Interface Module |
| BCM | Body Control Module |
| DDM | Driver Door Module |
| DSM | Driver Seat Module |
| ECG | Enhanced Central Gateway |
| IPC | Instrument Panel Cluster |
| PDM | Passenger Door Module |
| SCMB\_PSM | Passenger Seat Module |
| SCMG / SCMH | Multicontour Seat Module |
| SOBDMC\_HPCM | Powertrain Control Module |
| RCM | Restraint Control Module |

Table 3‑2: Electrical Components

#### E/E Connections

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Connection Name | **Connection Type** | **Protocol**  Only if ‘Connection Type’ is “Network”/”RF-Digital” | **Description** | **Allocated Messages**  Only if ‘Connection Type’ is “Network”/”RF-Digital” | **Connected Nodes** |
| HS-CAN2 | Network | CAN (High Speed) | Infotainment High Speed CAN bus |  | … |
| HS-CAN3 | Network | CAN (High Speed) | Infotainment High Speed CAN bus |  | … |
| FD-CAN1 | Network | CAN FD | Flexible Data CAN bus |  | … |
| FD-CAN3 | Network | CAN FD | Flexible Data CAN bus |  | … |
| MS-CAN1 | Network | CAN (Mid Speed) | Infotainment Middle Speed CAN bus |  | … |

Table 3‑3: E/E Connections

#### Signal List

***#Hint:*** *Refer to the* [*Data Dictionary*](#_Data_Dictionary) *-* [*Technical Signals*](#_Technical_Signals)*.*

### Software Component Architecture

#### Description

## Function Deployment

### Deployment Variants

|  |  |  |
| --- | --- | --- |
| **Deployment Variant Name** | Variant Description | Variant Condition (optional) |
| No Variant Defined |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

#### Deployment

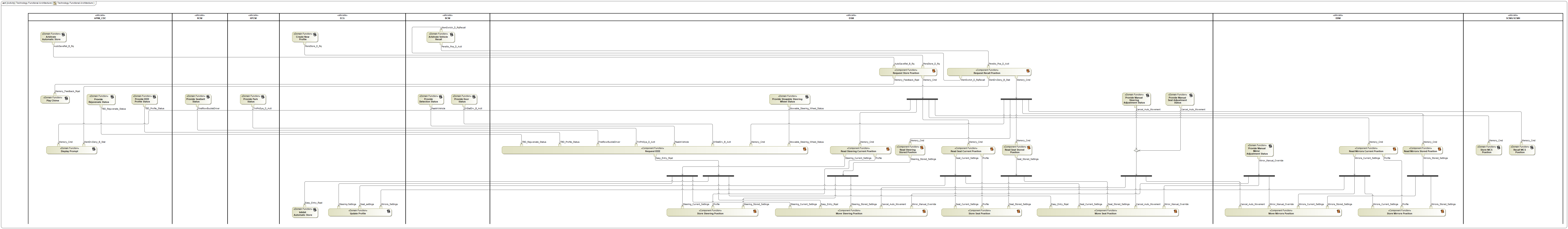
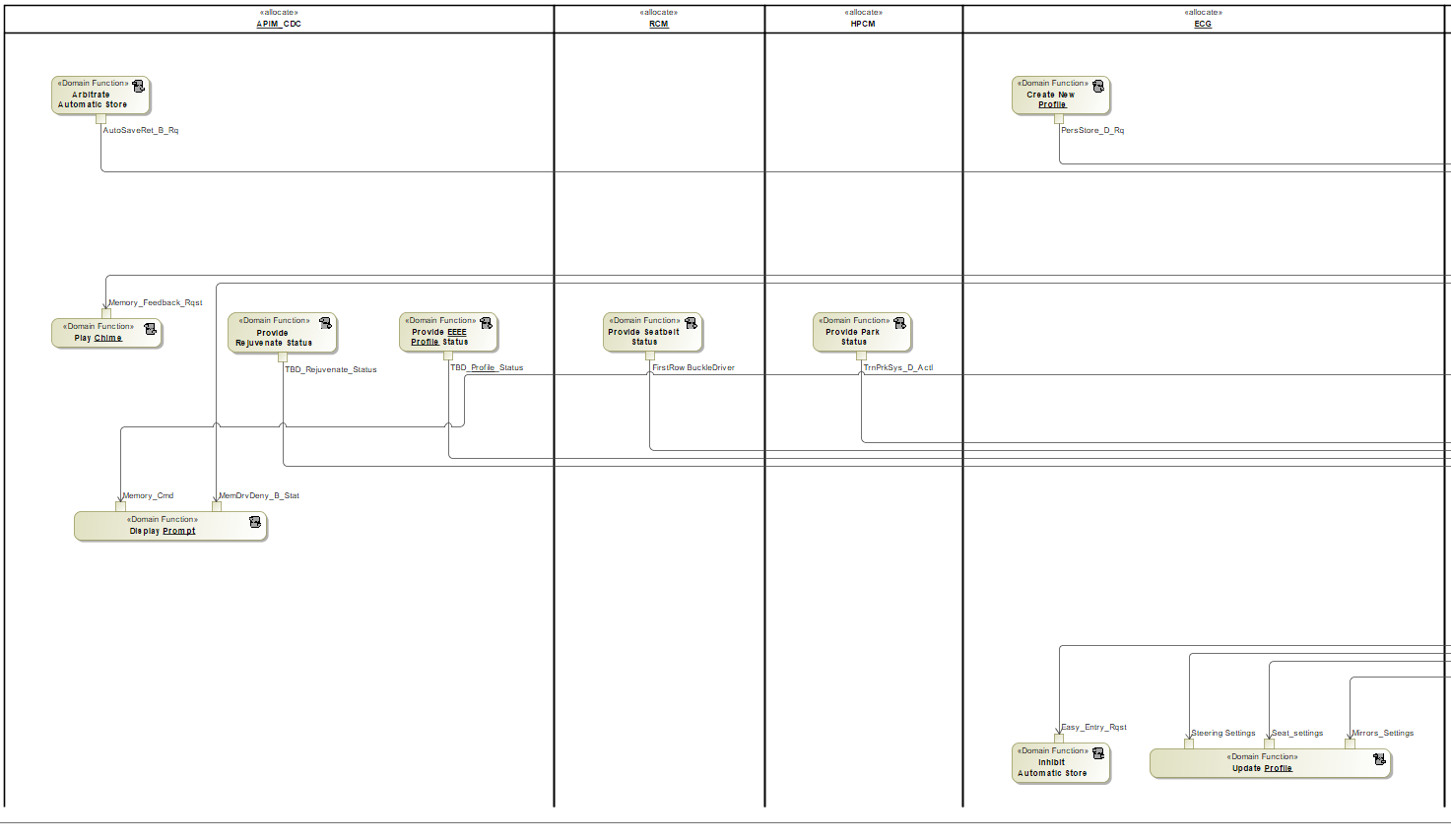
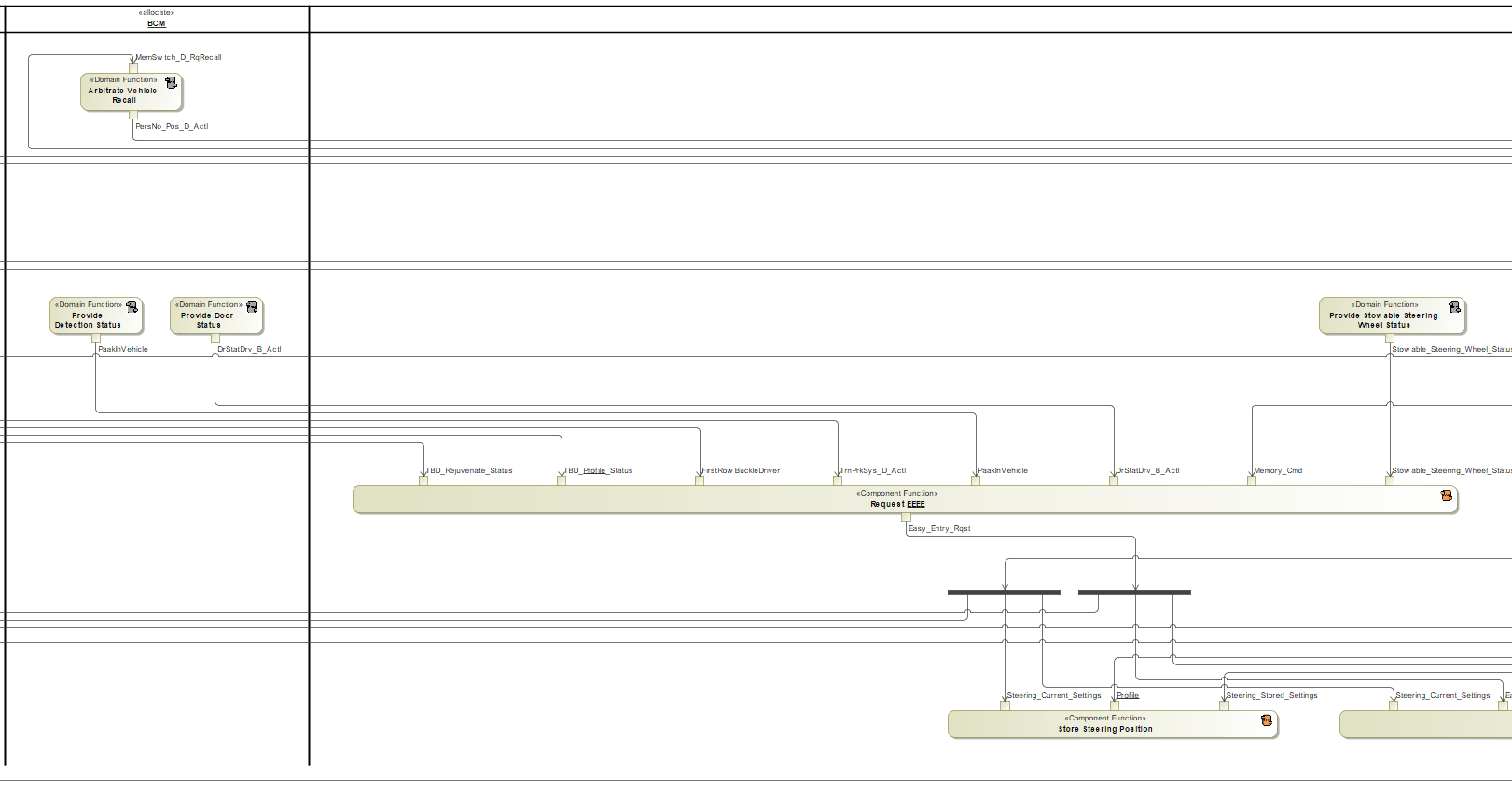
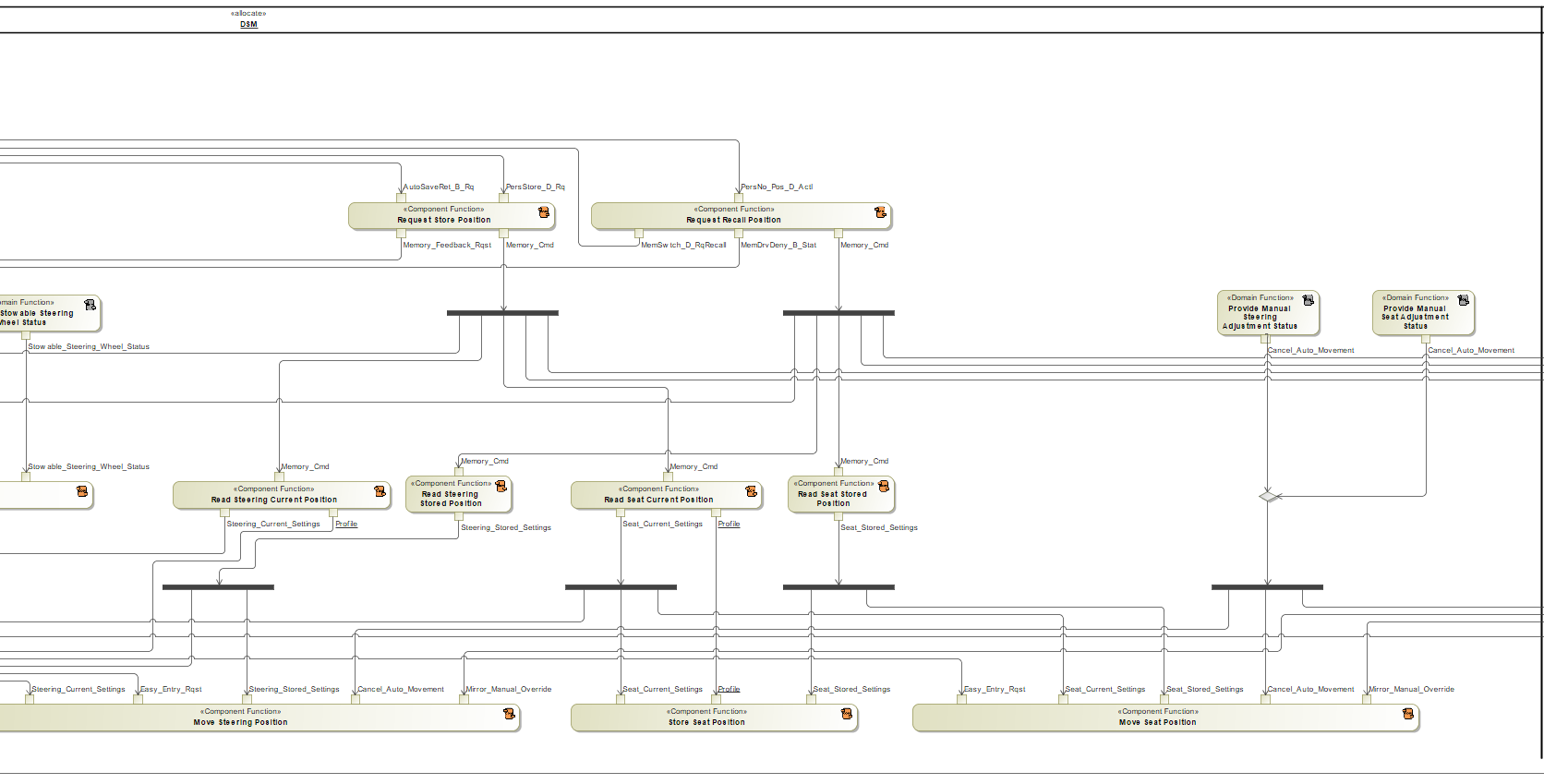


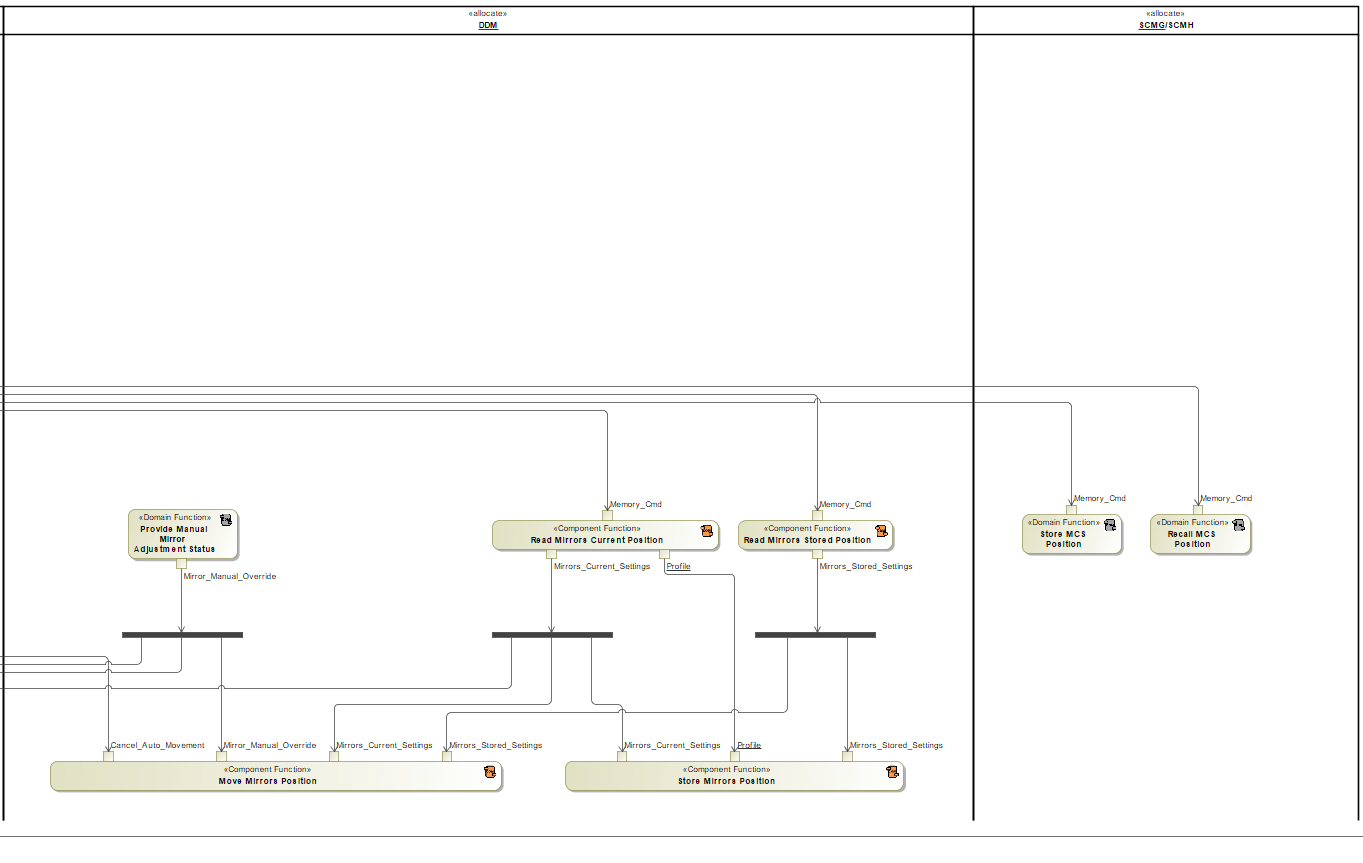
Figure 8: Vehicle System Behavior

Figure 8 enlarged in sections below:









### Function Allocation

| Component | Technology Function Name | Logical Function Name |
| --- | --- | --- |
|
| RCM |  |
| SCMB\_PSM |  |
| BCM |  |
| Classic Memory Physical |  |
| SCMG/SCMH |  |
| GWM\_ECG |  |
| ECM |  |
| DSM | Read Steering Stored Position | Read Steering Stored Position |
| Request Recall Position | Request Recall Position |
| Move Seat Position | Move Seat Position |
| Read Steering Current Position | Read Steering Current Position |
| Store Seat Position | Store Seat Position |
| Move Steering Position | Move Steering Position |
| Store Steering Position | Store Steering Position |
| Read Seat Stored Position | Read Seat Stored Position |
| Read Seat Current Position | Read Seat Current Position |
| Request Store Position | Request Store Position |
| Request EEEE | Request EEEE |
|  |
| HUD |  |
| GFM2\_SCSM |  |
| SOBDMC\_HPCM |  |
| IPC |  |
| APIM |  |
| PDM |  |
| DDM | Read Mirrors Current Position | Read Mirrors Current Position |
| Read Mirrors Stored Position | Read Mirrors Stored Position |
| Move Mirrors Position | Move Mirrors Position |
| Store Mirrors Position | Store Mirrors Position |
|  |
| ECG |  |
| HPCM |  |

Table 3‑5: Function Allocation Table (Basic)

# Feature Implementation Modeling

## Component Interaction Diagrams

### Scenario: “System Startup / Shutdown”

### Scenario: “Normal Operation”

## Component Interface Behavior Diagrams

# Feature Implementation Requirements

## Functional Safety

### ASIL Decomposition of Technical Safety Requirements

*Not Applicable*

## Requirements on Components

### ECG

### RCM

### DSM

#### Technology Function -2120879973.png **Read Steering Stored Position**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Memory\_Command | Memory\_Cmd | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Read Steering Stored Position

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | | **Technical Signal Name** | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Sored\_Steering\_Setting | | n/a | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | 6.0 | | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Read Steering Stored Position

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  |  |  |  |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

|  |  |  |
| --- | --- | --- |
| 5 | Classic Memory Functions - Steering | The DSM shall implement the following functions for steering: - Read Steering Current Position - Read Steering Stored Position - Move Steering Position - Store Steering Position |

#### Technology Function -2120879973.png **Request Recall Position**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| EM\_Recall | PersNo\_Pos\_D\_Actl | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Request Recall Position

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | | **Technical Signal Name** | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Memory\_Command | | Memory\_Cmd | |  |  |  |
| Deny\_Recall | | Mem\_DrvDeny\_B\_Stat | |  |  |  |
| CM\_Recall | | MemSwtch\_D\_RqRecall | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | 6.0 | | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Request Recall Position

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  |  |  |  |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

|  |  |  |
| --- | --- | --- |
| 6 | Classic Memory Function - Recall | The DSM shall implement the following functions for Classic Memory recalls: - Request Recall Position |

#### Technology Function -2120879973.png **Move Seat Position**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Manual\_Adjustment | Mirror\_Manual\_Override | | |  |  |  |
| Manual\_Adjustment | Cancel\_Auto\_Movement | | |  |  |  |
| EEEE\_Request\_Seat | Easy\_Entry\_Rqst | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Move Seat Position

###### Outputs

(No outputs have been defined)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
|  |  |  |  |  |

Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  |  |  |  |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

|  |  |  |
| --- | --- | --- |
| 4 | Classic Memory Functions - Seats | The DSM shall implement the following functions for seats: - Read Seat Current Position - Read Seat Stored Position - Move Seat Position - Store Seat Position |

#### Technology Function -2120879973.png **Read Steering Current Position**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Memory\_Command | Memory\_Cmd | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Read Steering Current Position

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | | **Technical Signal Name** | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Current\_Steering\_Settings | | **TBD** | |  |  |  |
| Profile | | n/a | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | 6.0 | | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Read Steering Current Position

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
| 39 | Personalization – Memory Store and Personal and Portable Profile | Added |  | Personal and Portable Profile (PPP) needs the updated settings. These are stored in a redundant profile and sent to the cloud. |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

39 Personalization - Memory Store and Personal Portable Profile

"When the following functions receive 'Memory\_Command' as 'Store' and the current position then the following functions shall output the profile and position to be stored to the 'Update Profile with Positional Settings' function

- Read Mirrors Current Position

- Read Seat Current Position

- Read Steering Current Position

- Read Pedals Current Position

- Read AHUD Current Brightness"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: 39 | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * -657675114.png Feature\_Req\_22 Personalization - Memory Store | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

|  |  |  |
| --- | --- | --- |
| 5 | Classic Memory Functions - Steering | The DSM shall implement the following functions for steering: - Read Steering Current Position - Read Steering Stored Position - Move Steering Position - Store Steering Position |

#### Technology Function -2120879973.png **Store Seat Position**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Profile | n/a | | |  |  |  |
| Current\_Seat\_Setting | n/a | | |  |  |  |
| Stored\_Seat\_Setting | n/a | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Store Seat Position

###### Outputs

(No outputs have been defined)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
|  |  |  |  |  |

Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  |  |  |  |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

|  |  |  |
| --- | --- | --- |
| 4 | Classic Memory Functions - Seats | The DSM shall implement the following functions for seats: - Read Seat Current Position - Read Seat Stored Position - Move Seat Position - Store Seat Position |

#### Technology Function -2120879973.png **Move Steering Position**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | | **Technical Signal Name** | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Current\_Seat\_Setting | | n/a | |  |  |  |
| Stored\_Seat\_Setting | | n/a | |  |  |  |
| Manual\_Adjustment | | Mirror\_Manual\_Override | |  |  |  |
| Manual\_Adjustment | | Cancel\_Auto\_Movement | |  |  |  |
| EEEE\_Request\_Steering | | Easy\_Entry\_Rqst | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | 6.0 | | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Move Steering Position

###### Outputs

(No outputs have been defined)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
|  |  |  |  |  |

Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  |  |  |  |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

|  |  |  |
| --- | --- | --- |
| 5 | Classic Memory Functions - Steering | The DSM shall implement the following functions for steering: - Read Steering Current Position - Read Steering Stored Position - Move Steering Position - Store Steering Position |

#### Technology Function -2120879973.png **Store Steering Position**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | | **Technical Signal Name** | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Profile | | n/a | |  |  |  |
| Current\_Steering\_Setting | | n/a | |  |  |  |
| Stored\_Steering\_Setting | | n/a | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | 6.0 | | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Store Steering Position

###### Outputs

(No outputs have been defined)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
|  |  |  |  |  |

Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  |  |  |  |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

|  |  |  |
| --- | --- | --- |
| 5 | Classic Memory Functions - Steering | The DSM shall implement the following functions for steering: - Read Steering Current Position - Read Steering Stored Position - Move Steering Position - Store Steering Position |

#### Technology Function -2120879973.png **Read Seat Stored Position**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Memory\_Command | Memory\_Cmd | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Read Seat Stored Position

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Stored\_Seat\_Setting | n/a | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Read Seat Stored Position

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  |  |  |  |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

|  |  |  |
| --- | --- | --- |
| 4 | Classic Memory Functions - Seats | The DSM shall implement the following functions for seats: - Read Seat Current Position - Read Seat Stored Position - Move Seat Position - Store Seat Position |

#### Technology Function -2120879973.png **Read Seat Current Position**

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Memory\_Command | Memory\_Cmd | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Read Seat Current Position

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Profile | n/a | | |  |  |  |
| Current\_Seat\_Settings | **TBD** | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Read Seat Current Position

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
| 39 | Personalization – Memory Store and Personal and Portable Profile | Added |  | Personal and Portable Profile (PPP) needs the updated settings. These are stored in a redundant profile and sent to the cloud. |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

39 Personalization - Memory Store and Personal Portable Profile

"When the following functions receive 'Memory\_Command' as 'Store' and the current position then the following functions shall output the profile and position to be stored to the 'Update Profile with Positional Settings' function

- Read Mirrors Current Position

- Read Seat Current Position

- Read Steering Current Position

- Read Pedals Current Position

- Read AHUD Current Brightness"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: 39 | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * -657675114.png Feature\_Req\_22 Personalization - Memory Store | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

|  |  |  |
| --- | --- | --- |
| 4 | Classic Memory Functions - Seats | The DSM shall implement the following functions for seats: - Read Seat Current Position - Read Seat Stored Position - Move Seat Position - Store Seat Position |

#### Technology Function -2120879973.png **Request Store Position**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Request Store | AutoSaveRet\_B\_Rq | | |  |  |  |
| EM\_Recall | PersStore\_D\_Rq | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Request Store Position

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Memory\_Command | Memory\_Cmd | | |  |  |  |
| Chime Status | Memory\_Feedback\_Rqst | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Request Store Position

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
| 57 | Auto Save Feature Store 1 | Added |  | Auto Save is a new input to trigger a Classic Memory store |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

57 Auto Save Feature Store 1

When the 'Request Store Position' function receives 'Auto\_Save\_Store' as 'Store' then the 'Request Store Position' function shall trigger a Classic Memory store for the current profile

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: 57 | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * -2113689296.png Feature\_Req\_30 Auto Save Feature Store | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

|  |  |  |
| --- | --- | --- |
| 7 | Classic Memory Function - Store | The DSM shall implement the following functions for Classic Memory stores: - Request Store Position |

#### Technology Function -2120879973.png **Request EEEE**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| TBD\_Profile\_Status | TBD\_Profile\_Status | | |  |  |  |
| Seatbelt Status | FirstRowBuckleDriver | | |  |  |  |
| Park Status | TrnPrkSys\_D\_Actl | | |  |  |  |
| Door\_Status | DrStatDrv\_B\_Actl | | |  |  |  |
| Detection\_Status | PaakInVehicle | | |  |  |  |
| Rejuvenate\_Status | **TBD** | | |  |  |  |
| Memory\_Command | Memory\_Cmd | | |  |  |  |
| Stowable\_Steering\_Wheel\_Status | n/a | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Request EEEE

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | | **Technical Signal Name** | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| EEEE\_Request\_Seat | | Easy\_Entry\_Rqst | |  |  |  |
| EEEE\_Request\_Steering | | Easy\_Entry\_Rqst | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | 6.0 | | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Request EEEE

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| EE Seat Horizontal Offset | Horiz\_EE\_Offset\_Cfg |  | Choose an item. | Easy Entry Easy Exit - |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
| 73 | Easy Entry Easy Exit - Drive Control Optimization - Offset Position - Park 1 | Added |  | Drive Control Optimization (DCO) drives need for new input to trigger Easy Entry Easy Exit (EEEE) |
| 74 | Easy Entry Easy Exit - Drive Control Optimization - Offset Position - SeatBelt 1 | Added |  | “ “ |
| 75 | Easy Entry Easy Exit - Drive Control Optimization - Drive Position 1 | Added |  | “ “ |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

73 Easy Entry Easy Exit - Drive Control Optimization - Offset Position - Park 1

"While

- 'Configuration' is 'DCO'

- Easy Entry Easy Exit is set Enabled for the active profile

- Easy Entry Easy Exit controlled commodities are in the drive position

- 'Vehicle Speed' is 'Slow'

- 'Gear' is 'Recall Available'

- 'Seatbelt Status' is 'Belted'

When the 'Request EEEE' function receives transition of 'Park Status' from 'Not Park' to 'Park' then the 'Request EEEE' function shall output 'EEEE\_Request\_Seat' and 'EEEE\_Request\_Steering' as 'EEEE'"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: 73 | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * -2113689296.png Feature\_Req\_47 Easy Entry Easy Exit - Drive Control Optimization - Easy Exit Offset Position - Park Transition | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

74 Easy Entry Easy Exit - Drive Control Optimization - Offset Position - Seat Belt 1

"While

- 'Configuration' is 'DCO'

- Easy Entry Easy Exit is set Enabled for the active profile

- Easy Entry Easy Exit controlled commodities are in the drive position

- 'Vehicle Speed' is 'Slow'

- 'Gear' is 'Recall Available'

- 'Park Status' is 'Park'

When the 'Request EEEE' function receives transition of 'Seatbelt Status' as 'Unbelted' to 'Belted' then the 'Request EEEE' function shall output 'EEEE\_Request\_Seat' and 'EEEE\_Request\_Steering' as 'EEEE'"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: 74 | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * -2113689296.png Feature\_Req\_48 Easy Entry Easy Exit - Drive Control Optimization - Easy Exit Offset Position - Seatbelt Transition | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

75 Easy Entry Easy Exit - Drive Control Optimization - Drive Position 1

"While

- 'Configuration' is 'DCO'

- Easy Entry Easy Exit is set Enabled for the active profile

- Easy Entry Easy Exit controlled commodities are in the Easy Entry Easy Exit offset position

- 'Vehicle Speed' is 'Slow'

- 'Gear' is 'Recall Available'

When the 'Request EEEE' function receives

- 'Detection Status' as 'Inside

- 'Door Status' as 'Closed'

Then the 'Request EEEE' function shall output 'EEEE\_Request\_Seat' and 'EEEE\_Request\_Steering' as 'Drive'"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: 75 | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * -2113689296.png Feature\_Req\_50 Easy Entry Easy Exit - Drive Control Optimization - Drive Position | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

|  |  |  |
| --- | --- | --- |
| 8 | Classic Memory Function - EEEE | The DSM shall implement the following functions for Classic Memory stores: - Request EEEE |

### HPCM

HPCM

### IPC

IPC

### APIM

APIM

### DDM

DDM

#### Technology Function 1045166107.png **Read Mirrors Current Position**

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Memory\_Command | Memory\_Cmd | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Read Mirrors Current Position

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | | **Technical Signal Name** | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Profile | | n/a | |  |  |  |
| Current\_Mirror\_Settings | | n/a | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | 6.0 | | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Read Mirrors Current Position

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
| 39 | Personalization – Memory Store and Personal and Portable Profile | Added |  | Personal and Portable Profile (PPP) needs the updated settings. These are stored in a redundant profile and sent to the cloud. |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

39 Personalization - Memory Store and Personal Portable Profile

"When the following functions receive 'Memory\_Command' as 'Store' and the current position then the following functions shall output the profile and position to be stored to the 'Update Profile with Positional Settings' function

- Read Mirrors Current Position

- Read Seat Current Position

- Read Steering Current Position

- Read Pedals Current Position

- Read AHUD Current Brightness"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: 39 | | | | | | | |
| **Rationale** |  | | | | | | |
| **Acceptance Criteria** |  | | | | | | |
| **Notes** |  | | | | | | |
| **Source** |  | | | | | **Owner** |  |
| **Source Req.** | * -657675114.png Feature\_Req\_22 Personalization - Memory Store | | | | | **V&V Method** |  |
| **Type** |  | | | **Priority** |  | **Status** |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | | |

|  |  |  |
| --- | --- | --- |
| 3 | Classic Memory Functions - Mirrors | The DDM shall implement the following functions: - Read Mirrors Current Position - Read Mirrors Stored Position - Move Mirrors Position - Store Mirrors Position |

#### Technology Function 1045166107.png **Read Mirrors Stored Position**

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | **Technical Signal Name** | | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Memory\_Command | Memory\_Cmd | | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | | 6.0 | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Read Mirrors Stored Position

###### Outputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | | **Technical Signal Name** | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Stored\_Mirror\_Settings | | n/a | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | 6.0 | | End of Requirement | | | |

Table 5‑3: Output Signal mappings of Function Read Mirrors Stored Position

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  |  |  |  |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

|  |  |  |
| --- | --- | --- |
| 3 | Classic Memory Functions - Mirrors | The DDM shall implement the following functions: - Read Mirrors Current Position - Read Mirrors Stored Position - Move Mirrors Position - Store Mirrors Position |

#### Technology Function 1045166107.png **Move Mirrors Position**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | | **Technical Signal Name** | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Stored\_Mirrors\_Settings | | n/a | |  |  |  |
| Current\_Mirrors\_Settings | | n/a | |  |  |  |
| Manual\_Adjustment | | Mirror\_Manual\_Override | |  |  |  |
| Manual\_Adjustment | | Cancel\_Auto\_Movement | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | 6.0 | | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Move Mirrors Position

###### Outputs

(No outputs have been defined)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
|  |  |  |  |  |

Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID**  (of Logical Function) | **Requirement Title** | **Modification** | **Requirement ID**  (of Technology Function) | **Comment** |
|  |  |  |  |  |

Table 5‑5: Component Specific Requirements

###### Component Specific Requirements

|  |  |  |
| --- | --- | --- |
| 3 | Classic Memory Functions - Mirrors | The DDM shall implement the following functions: - Read Mirrors Current Position - Read Mirrors Stored Position - Move Mirrors Position - Store Mirrors Position |

#### Technology Function 1045166107.png **Store Mirrors Position**

##### Function Interfaces

###### Inputs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID: | | | | | | |
| **Logical Signal Name** | | **Technical Signal Name** | | **Mapping Details** *(Conditional)* | **Subscriber Interface** | **Connection**  (*Optional)* |
| Profile | | n/a | |  |  |  |
| Stored\_Mirrors\_Settings | | n/a | |  |  |  |
| Current\_Mirrors\_Settings | | n/a | |  |  |  |
| [Req. Template](http://wiki.ford.com/display/RequirementsEngineering/Requirements+Attributes) Version | 6.0 | | End of Requirement | | | |

Table 5‑2: Input Signal mappings of Function Store Mirrors Position

###### Outputs

(No outputs have been defined)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Signal Name** | **Technical Signal Name** | **Mapping Details**  *(Conditional)* | **Publisher Interface** | **Connection**  *(Optional)* |
|  |  |  |  |  |

Table 5‑3: Output Signal mappings of Function “MyLogicalFunctionA\_Component1”A

###### Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Parameter Name** | **Technical Parameter Name** | **Mapping Details** *(Conditional)* | **Method** | **Method Details** |
| Name should be a Word reference to the “*Logical Parameters*” name bookmark in the Data Dictionary | Name should be a Word reference to the “*Technical Parameters*” name bookmark in the Data Dictionary | If mapping is not 1:1 you might reference a Mapping description object from the *Mappings* section | Choose an item. | Depends on Method selection. For Method 2 a DID including start bit and length could be given. For Central Car Config a signal could be referenced |
|  |  |  |  |  |

Table 5‑4: Parameter mappings of Function “MyLogicalFunctionA\_Component1”

###### Interface Requirements

##### Function Requirements

###### Component Specific Requirements

|  |  |  |
| --- | --- | --- |
| 3 | Classic Memory Functions - Mirrors | The DDM shall implement the following functions: - Read Mirrors Current Position - Read Mirrors Stored Position - Move Mirrors Position - Store Mirrors Position |

### SCMG/SCMH

SCMG/SCMH

### BCM

BCM

### PDM

PDM

## Requirements on Connections

### Networks

#### “CAN Bus xxx”

*Not Applicable*

##### Protocol Requirements

*Not Applicable*

##### Electrical Requirements

*Not Applicable*

#### “LIN Bus xxx”

*Not Applicable*

##### Protocol Requirements

###### Schedule Table

*Not Applicable*

##### Electrical Requirements

*Not Applicable*

#### “Ethernet xxx”

*Not Applicable*

### HW I/Os

*Not Applicable*

#### “HW I/O xxx”

## Requirements on Development Process

|  |  |
| --- | --- |
| Classic Memory Storage | The Classic Memory feature shall store settings in non-volatile memory |
| Driver Seat Accuracy | Power Driver Seat shall recall with accuracy in accordance with Seat Team strategy, see requirement ST-0037 |

# Open Concerns

***#Hint:*** *The following list presents known issues that have to be discussed or clarified over the course of the on-going requirements engineering.*

| ID | Concern Description | e-Tracker Reference | Status | Solution |
| --- | --- | --- | --- | --- |
| 1 | DCO   * Never specified if DCO notifies features via signal or configuration * May be implemented OTA after Job 1 |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |

Table 6‑1: Open Concerns

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Date | Description | Approved by | Responsible |
| A |  | Initial version |  | Jbaden1 |
|  |  |  |  |  |

## Template Revisions

*#Important: Do not change this section*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Rev. | Date | Description | Responsible |
| 0 | 2 | 2015-08-05 | * TOC corrected * Document Properties adapted to match needs of VBA macros | Awegman1 |
| 1 | 0 | 2015-11-16 | * Revision History moved to chapter 7 * Table-Styles removed | Awegman1 |
| 1 | 1 | 2016-03-02 | * Rework according to PCL example | Jbaden1 |
| 1 | 2 | 2016-03-22 | * V1.3: Footer formating corrected (Issue 19) * “Constraints” chapter renamed to “Input Requirements” (Issue 20) | Jbaden1 |
| 1 | 3 | 2016-04-20 | * Broken Wiki links repaired | Jbaden1 |
| 2 | 0 | 2016-05-23 | * Prepared for Specification\_Macros.dotm v2.0 * Additional explanations added to ch. 2.2 “Input Requirements” (ARL and SDS requirements often go here) | Jbaden1 |
| 2 | 1 | 2016-07-08 | * Template version added to footer | Jbaden1 |
| 2 | 2 | 2016-07-15 | * Sample SysML diagrams added * Data Dictionary reworked * Alignment with relevant sections in SRD templated | Jbaden1 |
| 3 | 0 | 2016-09-05 | * Lessons learned from IPRB incorporated | Jbaden1 |
| 4 | 0 | 2016-09-27 | * Alignment with QPIP Feature Function Ownership workstream. Platform Spec renamed to Feature Implementation Spec | Jbaden1 |
| 4 | 1 | 2016-11-04 | * Chapters “Purpose” and “Scope” reworked. | Jbaden1 |
| 4 | 1 | 2016-11-10 | * Subsection for “Logical Service Interfaces” added. | Jbaden1 |
| 5 | 0 | 2017-01-13 | * Meta data updated for specification macros, version 3.1 * SW Unit chapter removed for the time being * Green boxes added for user hints | Jbaden1 |
| 5 | 1 | 2017-01-18 | * Minor editorial changes (e.g. hyperlinks highlighted in comments) | Jbaden1 |
| 5 | 1b | 2017-01-20 | * Some editorial corrections * Substructure of old Network Communication (now Connections) moved to Requirements on Connections | Jbaden1 |
| 6 | 0 | 2018-07-24 | * CR53: * Add new cover sheet * Add disclaimer section * Add the following meta-data to the doc properties for the the new cover sheet   + DocGis1ItemNumber   + DocGis2Classification   + DocType   + DocStatus   + DocIssueDate   + DocReleaseDate * CR63: Update FuSa sharepoint references in templates | Jbaden1 |
| 6 | 0 | 2018-08-06 | * CR81: Incorporate lessons learned from System Service Spec pilot (Vehicle Speed) into AFS and FIS | Jbaden1 |
| 6 | 0 | 2018-09-28 | * Broken links to RE Wiki repaired | Jbaden1 |
| 6 | 0 | 2018-10-31 | * Minor corrections on cover sheet and in footer to be more GIS compliant and VSEM aligned * “Overview” and “Description” exchanged in headings (following common sense) | Jbaden1 |
| 6 | 0 | 2018-11-30 | * Update of Functional Safety sections after review by Functional Safety Team * Initial support for variant handling | Jbaden1 |
| 6 | 0 | 2018-12-01 | * Variant condition fields added consistently * Links updated | Jbaden1 |
| 6 | 0 | 2018-12-11 | * Variant condition fields removed from mapping/allocation tables * Mapping tables simplified * Explanatory text for “Variants” sections revised | Jbaden1 |
| 6 | 0a | 2019-01-04 | * Chapter heading “Inherited Function Requirements” removed. Corresponding table renamed to “Requirements not cascaded”. * E/E Connection table got another column for allocated messages * Naming conventions for Implemented Functions corrected (FncName\_CmpName instead of FncName\_on\_CmpName) * Editorial corrections on the cover sheet * Explanatory text added to “Ethernet” section in chapter “Requirements on Connections” * AIS templates updated. Linked to Wiki page | Jbaden1 |
| 6 | 0a | 2019-01-04 | * Minor restructuring in FuSa chapter – after aligning with ECU Functional Spec * Bugfix: table 13 renamed from FTTI table to FHT table, includes a bug fix: each FSR is allocated to only one ECU/component | Jbaden1 |
| 6 | 0b | 2019-02-04 | * Change: Chapter “Interface Requirements” added to “Implemented Function xxx” section (to have a single chapter for to collect subscriber/publisher interface and mapping requirements which to not conform to the corresponding Data Dictionary objects) * Change: “CAN Interface” subsection renamed to “AIS Interfaces” again. Although several Subscriber/Publisher interface attributes are probably CAN bus specific, other attributes seem to be well suited for other networks than CAN. * Change: Chapter “ECU Specific Requirements” renamed to “Component Specific Requirements” in chapter “Implemented Function xxx”. Table “Requirements not cascaded” renamed to “Component Specific Requirements” and refined to describe changes from Logical Function requirements set more formally. This is also to help during VSEM import to identify those requirements of the Logical Function which cannot be simply carried over to the ECU. * Change: Explanatory text in section “Implemented Function xxx” improved. | Jbaden1 |
| 6 | 0c | 2019-02-05 | * Change: Layout of AIS Interfaces in Data Dictionary reworked to enable Excel Import | Jbaden1 |
| 6 | 0c | 2019-02-20 | * Bugfix: In AIS Interfaces none-picklist fields formatted as invisible | Jbaden1 |
| 6 | 1a | 2019-02-05 | Functional Safety related changes:   * Table “Architectural Redundancy Summary” updated * Section “Functional Flows for FTTI ‘xyz’” added to chapter “Component Interaction Diagrams” * Fault Tolerant Time Summary section added to Functional Safety chapter * Chapter “HW Metrics” added | Jbaden1 |
| 6 | 1a | 2019-04-02 | Headings of “Architectural Redundancy Summary” table clarified | Jbaden1 |
| 6 | 1a | 2019-04-10 | * ASIL Decomposition table moved from Function Spec into the Feature Implementation Spec (ASIL Decomposition of Technical Safety Requirements) * 2 alternative versions of the Function Allocation Table (Standard variant vs. Functional Safety variant) placed next to each other. | Jbaden1 |
| 6 | 1a | 2019-05-31 | * Function Allocation Table split into a base (non FuSa) part and a FuSa part to allow a more flexible mapping of MBSE functions (Logical and Technology) to RE functions (Atomic Logical and Implemented). | Jbaden1 |
| 6 | 1a | 2019-05-31 | * “Input Requirement” section reworked (symmetrically to all other templates). * Sections “Functional Flows for FTTI xyz” and “Fault Tolerant Time Summary” removed, because guidance is not available yet. * “Reference” and “Glossary” section moved back to introduction, i.e., to the very beginning of the document (such that also section 2 can already rely on it). * Some mostly editorial changes per request from FuSa team. | Jbaden1 |
| 6 | 1a | 2019-07-02 | * "Important" box added on cover sheet which points to the macros * “Input Requirements” section renamed to Input Information (after discussion with FuSa team) | Jbaden1 |
| 6 | 1a | 2019-07-17 | * Chapter “Message List” removed from CAN and LIN specific chapters of section “Requirements on Connections” | Jbaden1 |
| 6 | 1a | 2019-10-08 | * Chapter “ASIL Decomposition of Technical Safety Requirements”: Input TSRs are specified in the chapter right above the decomposition table. | Jbaden1 |
| 6 | 1a | 2019-10-09 | * Chapter “Service Oriented Communication” moved to section “Messages” in the Data Dictionary. Details from Central SW Wiki about FNV2 SOA added | Jbaden1 |
| 6 | 1a | 2019-10-25 | * Minor updates for HW IOs/Signals * Subsection “Functional Safety” removed from chapter “Feature Implementation Modeling”. Per requrest from FuSa team since no guidance is available how to model e.g. FHT timing diagram. | Jbaden1 |
| 6 | 1a | 2019-05-11 | * Copyright notice shortened and moved to cover sheet and added to footer (to be compliant [with Ford copyright guidelines](http://www.fgti.ford.com/client/NewFGTI/CopyrightNotice.html)) * Term “Disclaimer” no longer used for what is actually only a copyright notice | Jbaden1 |
| 6 | 1a | 2019-22-11 | * Some minor modifications for the SOA APIs/MQTT Messages in the section “Messages” of the Data Dictionary (section references Service Contracts via the API name) * Some minor updates of the Input/Output mapping tables in section “Requirements on Components” for mappings to SOA APIs and EDAS signals. | Jbaden1 |
| 6 | 1a | 2019-12-05 | * Upstream Documents section added to “Input Requirements/Documents” table * Custom style table formatting removed | Jbaden1 |
| 6 | 1a | 2020-01-07 | * Some fine tuning for naming conventions of E/E components and connections. * List of HW I/O signal types reduced to RF-A, RF-D, D, A, Networked and PWM. * Protocol column added to the E/E connection table | Jbaden1 |
| 6 | 1a | 2020-01-07 | * “HW Metric” and “Architecture Redundancy Summary” sections removed per request from the Functional Architecture Team (based on Governance Board decision [FSTGB-97](mailto:TrackLite%20%23%20FSTGB-97:%20https://www.tracklite.ford.com/prweb/PRAuth/TrackLiteSSO?pyActivity=@baseclass.RedirectAndRunWraper&ThreadName=WorkLinkThread&bPurgeTargetThread=true&AccessGroupName=FSTGB:ProjectAdministrators&Location=pyActivity%3DWork-.Open%26Action%3DReview%26HarnessPurpose%3DReview%26InsHandle%3DFORD-FSTGB-WORK+FSTGB-97)) * “Functional Safety” chapter moved to “Feature Implementation Requirements” section. “Function Allocation” chapter seemed no longer appropriate. | Jbaden1 |
| 6 | 1a | 2020-01-07 | * Ordering of fields in AIS interfaces tables modified to conform with the Macro Template and the Importer Sheet * Page Header: no longer in bold letters | Jbaden1 |
| 6 | 1a | 2020-03-09 | * Missing doc property “LatestSigMappingID” and “LatestAisInterfaceID” added * doc property “CopyrightDate” re-formatted to text and copyright date field in footer corrected * Version numbering re-initialized as 0.1 * Init value of version/revision date set to “yyyy/mm/dd” instead of “yyyy-mm-dd” to be in line with the “Edit Document Property” dialog * Type of “Latest….ID” doc properties changed from Text to Number | Jbaden1 |
| 6 | 1a | 2020-03-11 | * “Mapping” table removed from template. Has been migrated to macro. | Jbaden1 |
| 6 | 1a | 2020-03-13 | * Separate chapter “Technical Safety Requirements” removed. Content already covered by Allocation Table in chapter Function Allocation. * “Implemented Function” replaced by term “Technology Function” | Jbaden1 |

# Appendix

## Data Dictionary

### Logical Signals

**#Hint:** Logical Signals are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link**: [*RE Wiki – Adding a Logical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Logical+Signal+or+Parameter)

**#Macro**: Add Ins -> Add Requirement macro (select “Logical Signal” as type)

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Description** | **Details** |
| **Restriction Settings Status** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Cancel\_Auto\_Movement** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Seat Belt Status** | Indicates status of the seat belt | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Gear Status** | Indicates the vehicle status | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **FirstRowBuckleDriver** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Current Setting** | Current Position of the different commodities | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **HMI Request Recall** | Recall Request | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersNoPos\_D\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Chime Status** | Request chime to be audible | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Door Status** | Indicates status of Door | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Park Status** | Indicates if vehicle is in Park | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **SSW\_Active** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory Command** | Indicates Classic Memory request | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemSwtch\_D\_RqAssoc** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **EmPrflButtnAssoc\_D\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Prompt Status** | Request variant prompt to be displayed | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory Input Status** | Indicates user selected profile | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **TmPrkSys\_D\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key\_In\_Ignition\_Stat** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_Cmd** |  | Satisfies:  799476293.png Example Implementation Reqt  Source ECU:  Target ECU: |
| **MemSwtch\_D\_RqRecall** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_[x]\_SwPsngr\_Stat** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Easy\_Entry\_Rqst** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Switch Request Recall** | Recall Request | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Rejuvenate\_Active** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **DrStatDrv\_B\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request Current Position** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Stowable Steering Wheel Status** | Indicates the status of the Stowable Steering Wheel | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Pos\_D\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_Feedback\_Rqst** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request\_Store** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersNo\_D\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Mirror\_Manual\_Override** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Strategy Configuration** | Indicates status of Easy Entry Easy Exit | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Restriction Status** | Indicates status of Vehicle Mode | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **EEEE Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersRecallSrc\_D\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersStore\_D\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Personal Index Value** | Indicates user selected profile | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RecallEvent\_No\_Cnt** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemDvrDeny\_B\_Stat** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **CntrStk\_D\_RqRecall** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Vehicle Speed** | Indicates the speed of the vehicle | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **User Input** | User selects a memory button/profile | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_[x]\_Sw\_Stat** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Manual Adjustment** | Indicates if there has been a manual adjustment of any commodity setting | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **EEEE Profile Status** | Indicates if EEEE is enabled or disabled | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemStoreMsgTxt\_D\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemSwtchPsngrFdbck\_B\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Vehicle Speed** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key Detection Status** | Indicates the Status of the remote transmitter | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PaakInVehicle** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Button Association Mode** | Indicates status of Button Association Mode | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Pers{x}Key\_D\_Stat** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request Store** | Store Request | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key Status** | Indicates the Status of the Key | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |

### Logical Parameters

**#Hint:** Logical Parameters are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link**: [*RE Wiki – Adding a Logical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Logical+Signal+or+Parameter)

**#Macro:** Add Ins -> Add Requirement macro (select “Logical Parameter” as type)

### Technical Signals

**#Hint:** This section lists all GSDB + GDT + SW signals relevant for the feature deployment.

**#Link**: [*RE Wiki – Adding a Technical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Signal+or+Parameter)

**#Macro:** Add Ins -> Add Requirement macro (select “Technical Signal” as type)

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **Description** | **Details** |
| **Restriction Settings Status** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Cancel\_Auto\_Movement** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Seat Belt Status** | Indicates status of the seat belt | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Gear Status** | Indicates the vehicle status | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **FirstRowBuckleDriver** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Current Setting** | Current Position of the different commodities | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **HMI Request Recall** | Recall Request | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersNoPos\_D\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Chime Status** | Request chime to be audible | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Door Status** | Indicates status of Door | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Park Status** | Indicates if vehicle is in Park | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **SSW\_Active** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory Command** | Indicates Classic Memory request | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemSwtch\_D\_RqAssoc** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **EmPrflButtnAssoc\_D\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Prompt Status** | Request variant prompt to be displayed | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory Input Status** | Indicates user selected profile | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **TmPrkSys\_D\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key\_In\_Ignition\_Stat** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_Cmd** |  | Satisfies:  799476293.png Example Implementation Reqt  Source ECU:  Target ECU: |
| **MemSwtch\_D\_RqRecall** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_[x]\_SwPsngr\_Stat** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Easy\_Entry\_Rqst** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Switch Request Recall** | Recall Request | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Rejuvenate\_Active** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **DrStatDrv\_B\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request Current Position** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Stowable Steering Wheel Status** | Indicates the status of the Stowable Steering Wheel | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Pos\_D\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_Feedback\_Rqst** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request\_Store** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersNo\_D\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Mirror\_Manual\_Override** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Strategy Configuration** | Indicates status of Easy Entry Easy Exit | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Restriction Status** | Indicates status of Vehicle Mode | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **EEEE Request** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersRecallSrc\_D\_Actl** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PersStore\_D\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Personal Index Value** | Indicates user selected profile | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **RecallEvent\_No\_Cnt** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemDvrDeny\_B\_Stat** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **CntrStk\_D\_RqRecall** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Vehicle Speed** | Indicates the speed of the vehicle | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **User Input** | User selects a memory button/profile | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Memory\_[x]\_Sw\_Stat** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Manual Adjustment** | Indicates if there has been a manual adjustment of any commodity setting | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **EEEE Profile Status** | Indicates if EEEE is enabled or disabled | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemStoreMsgTxt\_D\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **MemSwtchPsngrFdbck\_B\_Rq** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Vehicle Speed** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key Detection Status** | Indicates the Status of the remote transmitter | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **PaakInVehicle** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Button Association Mode** | Indicates status of Button Association Mode | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Pers{x}Key\_D\_Stat** |  | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Request Store** | Store Request | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |
| **Key Status** | Indicates the Status of the Key | Satisfies:  *No reqs. satisfied*  Source ECU:  Target ECU: |

#### GSDB Signals

**#Hint:** This part of the Data Dictionary lists signals, which should go to the GSDB in VSEM, but do not exist in the GSDB in VSEM yet, but are or will be requested for the GSDB. Those would go temporarily to this section in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server) in VSEM.

#### HW I/Os

**#Hint:** This chapter lists signals, which will be mapped to hardwired I/Os. Those get typically refer to VSEM EDAS signals (or input/output signals of device transmittals in VSEM GDT).

#### Diagnostic Interfaces

**#Hint:** This chapter lists Diagnostic Interfaces (DTCs and DIDs), which get mapped to Logical Parameters in context of the Technology Functions in chapter “Parameters” of the Function Interfaces. Those DTC/DID names should match the names in the diagnostics specification (Part 2).

**#ToDo:** Currently the template below is just a proposal. A macro still needs to be created

##### DTCs

<Some Description of the DTC.

Refer to VSEM document “[Diagnostic Fault Coverage and DTC Numbers](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=yAUtrNhnx3NrTDAAAAAAAAAAAAA&servername=Production_Server)

[Design Consideration](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=yAUtrNhnx3NrTDAAAAAAAAAAAAA&servername=Production_Server)”, what to fill into the attributes below>

|  |  |
| --- | --- |
| **Test Period Time** |  |
| **Test Run Criteria,** |  |
| **Enable Criteria (EC)** |  |
| **Applicable** |  |
| **FailureTypeBytes** |  |
| **Test Period Time** |  |
| **Test Run Criteria,** |  |

##### DIDs

**#Hint**: This section lists diagnostic DID which Technical Parameters get mapped to.

**#Todo**: A proper template derived from the Part 2 spec still needs to be created.

### Technical Parameters

**#Hint:** This section lists all Method 2, Method 3 and calibration parameters relevant for the feature deployment.

**#Link**: [*RE Wiki – Adding a Technical Signal or Parameter*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Signal+or+Parameter)

**#Macro:** [Add Ins -> Add Requirement macro](http://wiki.ford.com/display/RequirementsEngineering/How+to+use+the+Specification+Templates#HowtousetheSpecificationTemplates-AddNewRequirement) (select “Technical Parameter” as type)

### Mappings

**#Hint**: This section lists mapping objects for Logical Signals / Parameters to their GSDB + GDT + SW counterparts (1:N mapping is supported). Mapping objects are managed in VSEM in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link:** [RE Wiki – Adding a Signal or Parameter Mapping](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Signal+or+Parameter+Mapping)

**#Macro:** Add Ins -> Add Requirement macro (select “Mapping” as type)

### Technical Interfaces

**#Hint:** This section lists port/interface details, which define how network/SW/HW signals are published / subscribed.

**#Link:** [*RE Wiki – Adding a Technical Interface*](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface)

#### AIS Interfaces

**#Hint:** This chapter lists the AIS subscriber and publisher interface objects (managed in VSEM), which are needed to deploy the feature to the E/E architecture. If AIS interfaces do not yet exist in VSEM, those may temporarily be managed as a workaround in the [*RE Data Dictionary*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=SoYl_k7px3NrTD&servername=Production_Server).

**#Link:** [System Engineering Portal – AIS Release 3.2](https://pd3.spt.ford.com/sites/fede/vsem-spls/Shared%20Documents/02-ais/methods/AIS%20Methods%20Document.pptx?web=1)  
[RE Wiki - AIS Interfaces](http://wiki.ford.com/display/RequirementsEngineering/Adding+a+Technical+Interface#AddingaTechnicalInterface-AisInterfaces)

[*Publisher Interface AIS in VSEM*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=zjYtY3Jcx3NrTDAAAAAAAAAAAAA&servername=Production_Server)

[*Subscriber Interface AIS in VSEM*](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=LSYtewY7x3NrTDAAAAAAAAAAAAA&servername=Production_Server)

**#Macro:** Add Ins -> Add Requirement macro (select “AIS Subscriber If” or “AIS Publisher If” as type)

##### Publisher Interfaces

##### Subscriber Interfaces

#### AUTOSAR Ports

**#Hint:** Those AUTOSAR Classic (provided and required) ports, which are used by the feature but are not managed in a central repository yet, could be listed here.

### Messages/APIs

#### CAN Bus “<Bus Name>”

**#Hint:** This section gives the relevant extract from the [Central Message Database (CMDB) in VSEM](https://www.vsemweb.ford.com/tc/launchapp?-attach=true&-s=226TCSession&-o=jXfpx2PHx3NrTDAAAAAAAAAAAAA&servername=Production_Server) .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CAN ID** | **Transmission Mode** | **Period** | **Signal Names** | **Transmitter(s)** | **Receiver(s)** |
|  |  |  |  |  |  |
|  |
|  |
|  |

#### LIN Bus “<Bus Name>”

#### AUTOSAR Interfaces

**#Hint:** Those AUTOSAR Classic (Sender/Receiver and Client/Server) Interfaces, which are used by the feature but not managed in a central repository yet, should be listed here.

#### SOA Service Contracts

**#Hint:** This part of the Data Dictionary lists Service APIs/MQTT messages and embedded data elements, which are used for the Service Oriented Architecture (SOA). If those APIs/MQTT messages already exist e.g. in the [*Central SW Service Catalog*](http://wiki.ford.com/display/CS/Service+Catalog), simply add a reference to those yet.

Information on FNV2 SOA can be found in the ECG wiki page

* MQTT Topic Naming: [*FNV2-SOA: MQTT Topic and Message Structure*](https://www.eesewiki.ford.com/display/ecg/FNV2-SOA%3A+MQTT+Topic+and+Message+Structure?src=sidebar)
* message syntax and proper naming can be found [*SOA API Messaging Guidelines*](https://www.eesewiki.ford.com/x/Q7rKAg)

For examples what to fill into the table fields below refer to [*Central SW Service Catalog*](http://wiki.ford.com/display/CS/Service+Catalog)

<Service contract purpose/behavior>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Messaging Pattern | Frequency  (For Data Broadcast Only) | Message Data Element(s)  (Must Match GPB) or applicable CAN signal | Description of Data Element(s) | Topic Name |
| Choose an item. |  | GBP Data element / CAN Signal name 1 | Detailed encoding of data element 1 |  |
| … |  |  |
| GBP Data element / CAN Signal name 1 | Detailed encoding of data element 3 |  |

### Encoding Types

**#Link:** [*RE Wiki – Adding Encoding Types*](http://wiki.ford.com/display/RequirementsEngineering/Adding+an+Encoding+Type)

**#Macro:** Add Ins -> Add Requirement macro (select “Encoding Type” as type)

Document ends here.